

Interpreting CP/M Error Messages: The Status Register

The Status Register monitors various conditions in the floppy disk controller (part of the circuitry of the AT8000's board). This register is made up of 8 bits, with each individual status bit having its own meaning.

When a CP/M error message appears on the screen, it may be followed by a number. This number refers to one or more of the status bits. [Although some of the status bits point out errors, not all do. Some simply tell which conditions in the floppy disk controller are active.] The Status Register chart below shows the individual bits (Bits 0 - 7). The decimal equivalent of each bit is also referenced because the SWP BIOS uses decimal output for error messages.

To use the following chart, first determine the type of command that was active when the error message occurred (whether the floppy disk controller was trying to seek, read, or write). Then cross reference this command with the decimal number that appeared in the error message.

For example, ERROR 64, when the system was trying to write to a disk, means that there is a write protect error. This means that you were attempting to write to a disk that was write protected (for example, if an 8" disk is involved, you need to place a write-protect tab on it, a 5 1/4" disk needs the tab removed).

If a number in an error message is not one of the 8 decimal numbers above (1, 2, 4, 8, 16, 32, 64, or 128), then it is a combination of two. For example, ERROR 24 means that there is a problem with Bit 4 (16 decimal) and with Bit 3 (8 decimal).

More detailed explanation of the bit meanings follow the chart.

A Summary of the Status Register

COMMAND ↓	STATUS BITS							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Decimal 128	Decimal 64	Decimal 32	Decimal 16	Decimal 8	Decimal 4	Decimal 2	Decimal 1
Seek	NOT READY	WRITE PROTECT	HEAD LOADED	SEEK ERROR	CRC ERROR	TRACK 0	INDEX	BUSY
Read	NOT READY	-	RECORD TYPE	RECORD NOT FOUND	CRC ERROR	LOST DATA	DATA REQUEST	BUSY
Write	NOT READY	WRITE PROTECT	WRITE FAULT	RECORD NOT FOUND	CRC ERROR	LOST DATA	DATA REQUEST	BUSY

NOT READY indicates that the requested drive is not ready. It can occur if the drive door is not closed, or if you inadvertently try to access a drive that doesn't exist.

WRITE PROTECT means that the write protect feature is active and the disk cannot be written to. To write to an 8" disk, you must place a foil write-protect tab on the disk. To write to a 5 1/4" disk, you must remove the write-protect tab from the disk.

HEAD LOADED means that the drive head is loaded and engaged.

RECORD TYPE. When a record is being read, this indicates the record-type code from the data field address mark. 1= deleted data mark. 0= normal data mark.

WRITE FAULT indicates that there is a fault in writing.

SEEK ERROR means that the desired track was not verified.

RECORD NOT FOUND means that the desired track, sector, or side was not found.

CRC ERROR means that there was a checksum error in data in a sector. If this occurs with a RECORD NOT FOUND or a SEEK error (decimal 24), then there is an error in one or more ID fields. If it occurs alone, it indicates error in a data field.

TRACK 0 indicates that the drive's Read/Write head is positioned to Track 0.

LOST DATA occurs when the computer did not respond to a data request in one byte time. This happens when you try to read a longer sector than the active program was expecting.

INDEX means that the index mark is detected from the drive.

DATA REQUEST is a copy of the data request output. It indicates that the data request is full on a read operation, or that the data request is empty on a write operation.

BUSY means that a command is under execution.

ATR8000 User Accessible Monitor Routines and Variables

This section gives the locations and calling sequences of user accessible I/O routines in the ATR8000 monitor. It also describes a number of important monitor variables that may need to be accessed by user written programs.

ATR8000 subroutines are accessed via a table of JUMP instructions beginning at memory location F000 hex. All monitor calls should be made to these entry points, since the actual addresses of the routines inside the ATR8000 will vary between different releases. Parameter passing conventions for the monitor fall into one of two groups. The character oriented I/O routines all pass data using the A and C registers while the disk routines pass parameters in a 9 byte long disk I/O parameter block pointed to by IX.

Storage for the monitor stack and working variables occupies the top 256 bytes of memory, from FFC0 to FFFF hex. Included are variables used by the monitor resident I/O drivers and interrupt service routines, some of which are described below. Programs should not attempt to write into any locations in this block that are not specifically mentioned below.

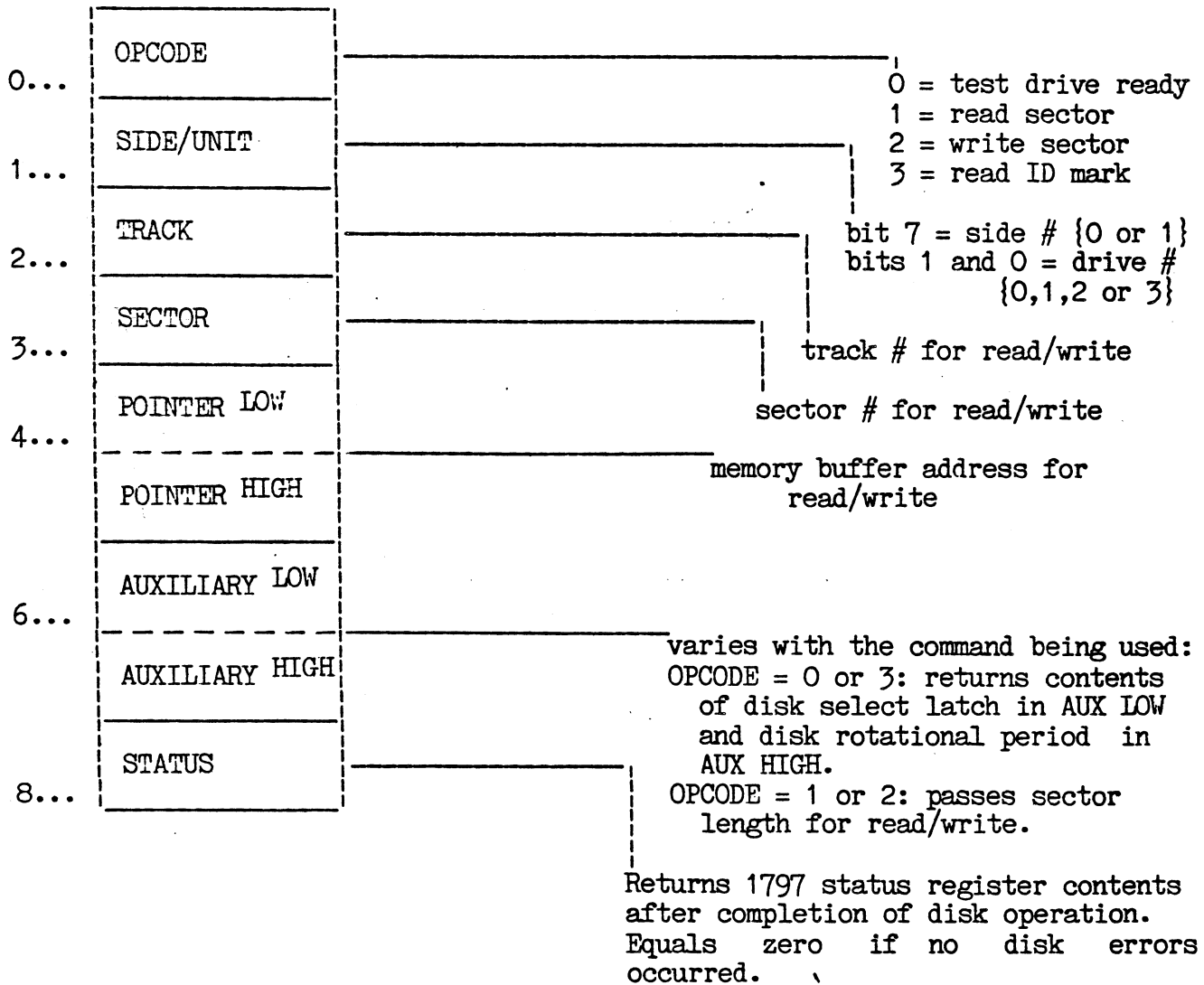
ATR8000 Subroutine Entry Points

F000	JP	COLD	;ATR8000 COLD START ENTRY
F003	JP	WARM	;ATR8000 WARM START ENTRY
F006	JP	CONST	;CONSOLE INPUT STATUS TEST
F009	JP	CONIN	;CONSOLE INPUT
F00C	JP	CONOUT	;CONSOLE OUTPUT
F00F	JP	DISKIO	;EXECUTE DISK OPERATION
F012	JP	LISTOUT	;LIST DEVICE OUTPUT
F015	JP	LISTST	;LIST DEVICE STATUS
F018	JP	REINIT	;CONSOLE REINITIALIZATION

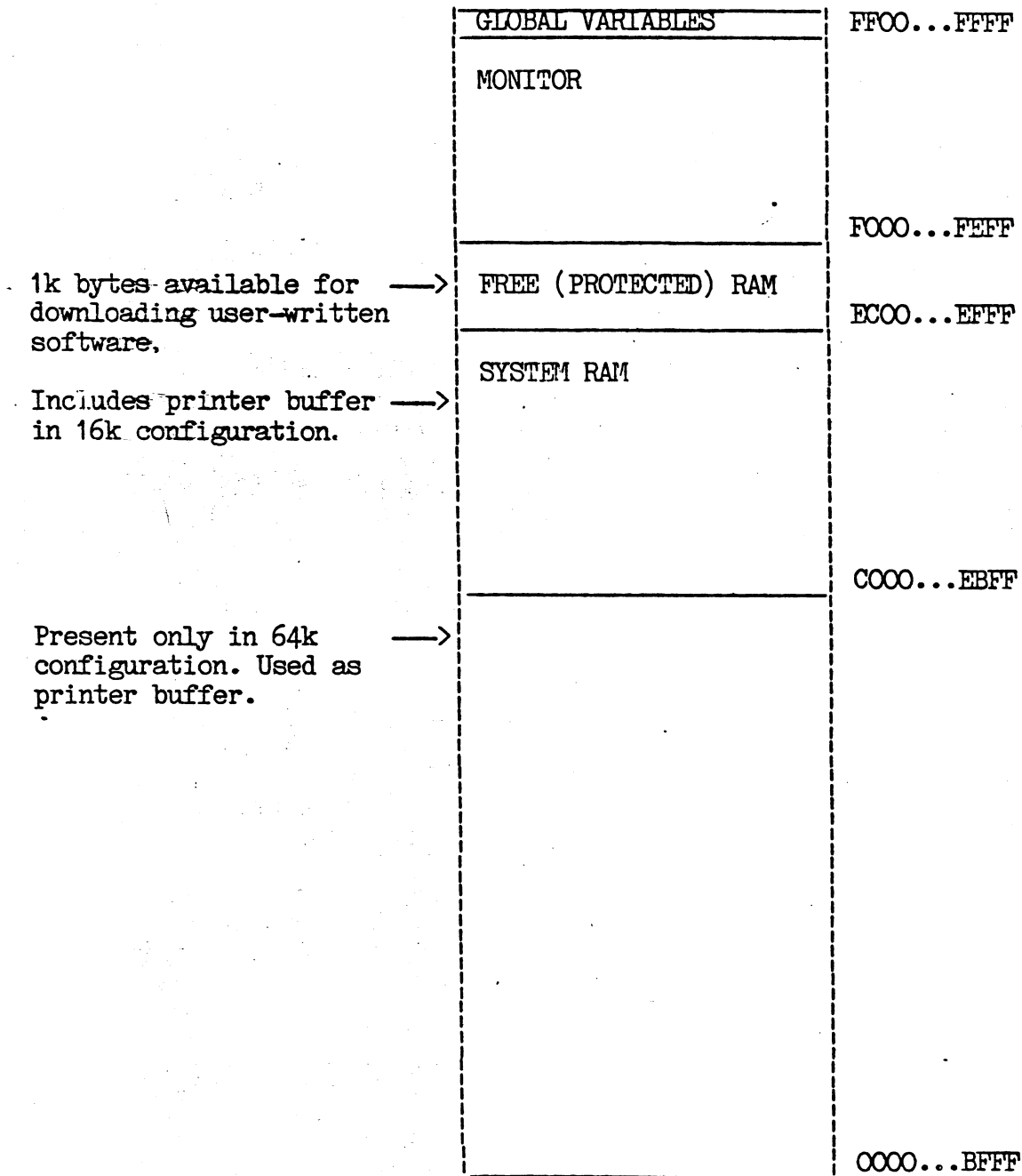
<u>Function</u>	<u>Parameters</u>	<u>Description</u>
COLD	IN: none OUT: does not return	Perform cold start initialization of ATR8000 monitor and enter command mode.
WARM	IN: none OUT: does not return	Enter ATR8000 monitor command mode with no reinitialization.
CONST	IN: none OUT: status in A	Test for data ready in console input FIFO and return status in A. If data is available A=0, else A=FF hex.
CONIN	IN: none OUT: character in A	Return character from console input FIFO in A. If FIFO is empty then loop until character is input.
CONOUT	IN: character in C OUT: none	Output character passed in C to the console output device.
DISKIO	IN: OUT: } see chart	Execute disk operation. See Disk I/O Control Block Format Chart on the following page.
LISTOUT	IN: character in C OUT: none	Output character to list device without checking for ready status.
LISTST	IN: none OUT: status in A	List device status. A=0 if printer is ready to print.
REINIT	IN: none OUT: none	Reinitialize console I/O after performing a disk operation.

Disk I/O Control Block Format

@IX



ATR8000 Memory Map



Appendix B: The ATR8000 Monitor

The ATR8000's monitor ROM can be accessed after the ATR8000 terminal program is run. (This terminal program is supplied with CP/M.) It can also be accessed from ATARI BASIC by typing in the ATARI BASIC program, ATRMON.BAS, listed below.

When the terminal program is run, the sign-on message #ATRMON appears on the screen. Four commands can be entered here:

Table 1: Monitor Commands

Command	Action
D<aaaa>	Dumps 128 bytes of memory on the screen (a block). <aaaa> represents the memory location. For example, D0 will dump the first 128 bytes of memory.
M<aaaa>	Command to examine and change ~~128 bytes of~~ memory.
G<aaaa>	Go to a particular location.
B	Boots CP/M when a CP/M disk is in Drive 1.

The ATR8000 monitor can be accessed from ATARI DOS or OS/A+ by entering and running ATRMON.BAS. The monitor commands in Table 1 also apply to this program.

ATRMON.BAS -- An ATARI BASIC Program

```

100 REM <<< ATRMON.BAS >>>
110 REM <<< D=DUMP MEM >>>
120 REM <<< M=EXAMINE MEM >>>
130 REM <<< G=EXICUTE AT ADD >>>
140 DIM X$(16)
150 X$="0123456789ABCDEF"
160 DIM C$(1)
170 DIM BUF$(256)
180 DIM H$(5)
190 REM SET UP USR Z
200 Z=ADR(" ")
210 DATA 104,76,89,228
220 FOR I=0 TO 3
230 READ X
240 POKE Z+I,X
250 NEXT I
260 REM PRESET DCB BUFFER
270 DCB=768
280 CMD=770:REM "R","W","G","S"
290 AX1=778:REM BYTE COUNT OR LOW ADD

```

APPENDIX B

```
300 AX2=779:REM HIGH ADD
310 DIR=771:REM SIO CONTROL
315 NUM=776:REM BYTE COUNT FOR SIO
320 IN=64:REM INPUT TO BUFFER
330 OUTB=128:REM OUTPUT WITH BUFFER
340 OUT=0:REM OUTPUT WITHOUT BUFFER
350 DATA 90,1,0,0,0,0,1,0,0,0,0,0
360 REM "2"=90,DEVICE I.D.
370 FOR I=0 TO 11
380 READ X
390 POKE DCB+I,X
400 NEXT I
410 REM SET UP BUFFER ADDRESS
420 A=ADR(BUF$)
430 X=INT(A/256)
440 POKE DCB+5,X
450 POKE DCB+4,A-X*256
460 INPUT H$
462 C$=H$(1,1)
465 H$=H$(2,LEN(H$))
470 IF ASC(C$)=ASC("D") THEN 600
480 IF ASC(C$)=ASC("M") THEN 1000
490 IF ASC(C$)=ASC("G") THEN 1500
500 GOTO 460
600 REM DUMP MEMORY
610 GOSUB 2000
620 GOSUB 5000
680 POKE CMD,ASC("R")
690 POKE DIR,IN
700 POKE AX1,128
705 POKE NUM,128
710 GOSUB 9000
720 N=0
725 M=D
730 FOR I=0 TO 127
740 IF N>0 THEN 760
745 D=M+I
747 GOSUB 4000
748 PRINT H$;
749 PRINT " ";
750 N=N+1
760 D=PEEK(A+I)
770 GOSUB 4000
780 PRINT H$(3,4);
790 PRINT " ";
800 N=N-1
801 IF N>0 THEN 820
802 FOR F=7 TO 0 STEP -1
803 G=PEEK((A+I)-F)
804 IF G<32 THEN 808
805 IF G>121 THEN 808
806 PRINT CHR$(G);
807 GOTO 809
808 PRINT " ";
809 NEXT F
815 PRINT
820 NEXT I
830 GOTO 460
```

```

1000 PRINT H$;
1010 PRINT " ";
1020 GOSUB 2000
1030 GOSUB 5000
1040 M=D
1050 POKE CMD,ASC("R")
1060 POKE DIR,IN
1070 POKE AX1,1
1080 POKE NUM,1
1090 GOSUB 9000
1100 D=PEEK(A)
1110 GOSUB 4000
1120 PRINT H$(3,4);
1130 INPUT H$
1140 IF ASC(H$)=ASC(".") THEN 460
1150 IF LEN(H$)=0 THEN 1190
1160 GOSUB 2000
1170 POKE A,D
1180 POKE CMD,ASC("W")
1190 POKE DIR,OUTB
1200 GOSUB 9000
1210 M=M+1
1220 D=M
1230 GOSUB 5000
1240 GOSUB 4000
1250 PRINT H$;
1260 PRINT " ";
1270 GOTO 1040
1280 PRINT H$
1290 GOSUB 2000
1300 GOSUB 5000
1310 POKE CMD,ASC("G")
1320 POKE DIR,OUT
1330 GOSUB 9000
1340 GOTO 460
2000 IF LEN(H$)<1 OR LEN(H$)>4 THEN 2099
2010 D=0
2020 FOR J=1 TO LEN(H$)
2030 K=ASC(H$(J,J))-ASC("0")
2040 IF K<0 THEN 2099
2050 IF K<10 THEN 2090
2060 K=K-7
2070 IF K<10 THEN 2099
2080 IF K>15 THEN 2099
2090 D=(D*16)+K
2095 NEXT J
2096 RETURN
2099 PRINT "INVALID HEX DATA"
2100 RETURN
4000 FOR J=4 TO 1 STEP -1
4010 K=D
4020 D=INT(D/16)
4030 H$(J,J)=X$(K-(D*16)+1)
4040 NEXT J
4050 RETURN
5000 Y=INT(D/256)
5010 POKE AX2,Y
5020 POKE AX1,D-Y*256

```

```

5030 POKE CMD,ASC("S")
5040 POKE DIR,OUT
5050 GOSUB 9000
5060 RETURN
9000 X=USR(Z)
9010 RETURN

```

10
 20
 30
 40
 50
 60
 70
 80
 90
 100
 110
 120
 130
 140
 150
 160
 170
 180
 190
 200
 210
 220
 230
 240
 250
 260
 270
 280
 290
 300
 310
 320
 330
 340
 350
 360
 370
 380
 390
 400
 410
 420
 430
 440
 450
 460
 470
 480
 490
 500
 510
 520
 530
 540
 550
 560
 570
 580
 590
 600
 610
 620
 630
 640
 650
 660
 670
 680
 690
 700
 710
 720
 730
 740
 750
 760
 770
 780
 790
 800
 810
 820
 830
 840
 850
 860
 870
 880
 890
 900
 910
 920
 930
 940
 950
 960
 970
 980
 990
 1000

—

APPENDIX B

```

C      ;      ... VARIABLES FOR DISK FORMAT FUNCTION ...
C      ;
C      FMTSTUF EQU      $
C      FRMPTR: DEFS      2      ;POINTER TO FORMAT DATA TABLE
C      SHWPTR: DEFS      2      ;POINTER TO SKEW TABLE
C      FILL:   DEFS      2      ;EMPTY SECTOR FILL DATA
C      NTRAKS: DEFS      1      ;NUMBER OF TRACKS
C      NSECTS: DEFS      1      ;NUMBER OF SECTORS
C      TRKSIZE: DEFS      2      ;TRACK LENGTH IN BYTES
C      0000A   C      FMTLEN EQU      $-FMTSTUF
C
C      FF69     C      SEQNUM: DEFS      2      ;TEMP SECTOR SEQUENCE NUMBER
C
C      FF6B     C      SEQPTR: DEFS      2      ;TEMP ERROR LOG TABLE POINTER
C      FF6D     C      TRKPTR: DEFS      2      ;POINTER TO START OF TRACK IMAGE
C      ;
C      ;      ... INITIALIZED LOCAL VARIABLES FOR DISK HANDLER ...
C      ;
C      CMBYTB: DEFB      0      ;COMMAND BYTE FOR READS/Writes
C      RWTRY:  DEFB      0      ;READ/WRITE RETRY COUNT
C      FNDTRY: DEFB      0      ;SEEK RETRY COUNT
C      ;
C      ;
C      IOBUFF EQU      0C000H   ;ATARI I/O BUFFER
C      0200     C      LEN      EQU      512
C      ;
C      C200     C      TRKBUF  EQU      IOBUFF+LEN   ;TRACK BUFFER FOR READS
C      ;
C      ;
C      .DEPHASE
C      ;
C      ;
C      0FD5     C      LAST      EQU      $
C      END

```


Global Variables for AT8000 2.0 ROM

```

C      ;*****
C      ;
C      ; GLOBAL VARIABLES FOR ATARI 280 ROM
C      ;
C      ;*****
C      ;
C      ;
C      ; RAM      EQU      $
C      ;
C      ;
C      ; KEYBUF:  DEFB      0,0,0,0      ;16 BYTE CONSOLE KEYBOARD INPUT FIFO
C      ;           DEFB      0,0,0,0
C      ;           DEFB      0,0,0,0
C      ;           DEFB      0,0,0,0
C      ;
C      ; CTCVEC:  DEFW      RETI          ;SPACE FOR 4 VECTORS FOR CTC
C      ;           DEFW      RETI
C      ;           DEFW      RETI
C      ;           DEFW      RETI
C      ;
C      ;           DEFW      PETI          ;SPACE FOR 4 EXTRA VECTORS
C      ;           DEFW      RETI
C      ;           DEFW      PETI
C      ;           DEFW      RETI
C      ;
C      ;
C      ;
C      ;
C      ; ... GLOBAL VARIABLES FOR PHYSICAL DISK HANDLER ...
C      ;
C      ; DRVTAB:  DEFB      255,255,255,255 ;HEAD POSITIONS FOR 4 DRIVES
C      ;           DEFB      0,0,0,0      ;DENSITY/TYPE/SELECT BITS FOR 4 DRIVES
C      ; RATES:   DEFB      16,16,16,16   ;SETTLING DELAYS / STEP RATES TABLE
C      ; UNIT:    DEFB      0             ;CURRENTLY SELECTED DISK#
C      ; TRACK:   DEFB      255          ;TRACK POSITION OF SELECTED DRIVE
C      ; DRVOFF:  DEFB      1            ;DRIVES-OFF FLAG FROM DISK TIMER IRQ
C      ; OUTCPY:  DEFB      00000000B    ;COPY OF DISK CONTROL LATCH
C      ; PERIOD:  DEFW      0             ;DISK SPIN PERIOD
C      ; HLDTIM:  DEFB      50           ;HEAD LOAD DELAY
C      ; RWMAX:   DEFB      10           ;MAX NUMBER OF READ/WRITE RETRIES
C      ;           DEFB      0,0,0,0     ;POOM FOR EXPANSION
C      ;
C      ;
C      ;
C      ; ... GLOBAL VARIABLES FOR ATARI HANDLER ...
C      ;
C      ; IDPTR:   DEFW      IDTAB          ;POINTER TO DEVICE ID TABLE
C      ; FSMVEC:  DEFW      PWRWAIT       ;POINTER FOR ATARI TASK STATE MACHINE
C      ; EXTVEC:  DEFW      DUMMY         ;POINTER FOR EXTRA TASK PROCESSOR
C      ; NEWLIN:  DEFB      2,CR,LF,0,0   ;PRINTER NEWLINE CHARACTERS
C      ;
C      ; PMSGC:   DEFB      0,0,10,0     ;PRINTER STATUS FRAME
C      ; PMASKS:  DEFB      10000000B    ;MASK ALL BITS BUT 'BUSY'
C      ;           DEFB      00000000B    ;COMPARE TO ZERO FOR READY
C      ; FMTPTR:  DEFW      FMTS          ;POINTER TO STANDARD FORMAT TABLES
C      ; FBASE:   DEFW      IDBUFF+(2*LEN) ;PUT PRINT BUFFER AFTER HERE
C      ; PSIZE:   DEFW      4095         ;MAX BUFFER INDEX OFFSET
C      ;
C      ; GLBSIZE  EQU      $-RAM          ;LENGTH OF GLOBAL VARIABLES
C      ;
C      ;
C      ; *** UNINITIALIZED SCRATCH VARIABLES COME AFTER HERE ***
C      ;
C      ; PCOUNT:  DEFW      0             ;BYTECOUNT FOR BUFFER
C      ; PINP:    DEFW      0             ;INPUT OFFSET
C      ; POUT:    DEFW      0             ;OUTPUT OFFSET
C      ; CMDFLG:  DEFB      0             ;COMMAND FRAME READY FLAG FROM IRQ
C      ; DMATRIX:
C      ;           DEFS      16           ;DRIVE POOP TABLES
C      ;           DEFS      16
C      ;           DEFS      16
C      ;           DEFS      16
C      ; OLDPTR:   DEFS      2             ;POINTER TO FIRST DRIVE ACCESSED
C      ; OKIOCB:  DEFS      16           ;DISK I/O COMMAND BLOCK
C      ; DRWCMD:  DEFS      1             ;R/W COMMAND FROM ATARI TO 'DISKIO'
C      ; LOGSIZ:  DEFS      2             ;LOGICAL SECTOR LENGTH FOR XFER
C      ; IDBUF:   DEFS      8             ;BUFFER FOR ID MARK READS
C      ; IDPTR:   DEFS      2             ;ATARI BLOCK INPUT POINTER
C      ; VFLAG:   DEFS      1             ;VERIFY FLAG FOR DISK WRITES
C      ;
C      ;

```

```

C      ;      ... VARIABLES FOR DISK FORMAT FUNCTION ...
C      ;
C      FMTSTUF EQU      8
C      FRMPTR: DEFS     2      ;POINTER TO FORMAT DATA TABLE
C      SKWPTR: DEFS     2      ;POINTER TO SKEW TABLE
C      FILL:  DEFS     2      ;EMPTY SECTOR FILL DATA
C      NTRAKS: DEFS     1      ;NUMBER OF TRACKS
C      NSECTS: DEFS     1      ;NUMBER OF SECTORS
C      TRKSIZ: DEFS     2      ;TRACK LENGTH IN BYTES

C      FMTLEN EQU      8-FMTSTUF

C      SEQNUM: DEFS     2      ;TEMP SECTOR SEQUENCE NUMBER
C      SEQPTR: DEFS     2      ;TEMP ERROR LOG TABLE POINTER
C      TRKPTR: DEFS     2      ;POINTER TO START OF TRACK IMAGE
C      ;
C      ;      ... LOCAL VARIABLES FOR DISK HANDLER ...
C      ;
C      CMDBYT: DEFB     0      ;COMMAND BYTE FOR READS/Writes
C      RWTRY:  DEFB     0      ;READ/WRITE RETRY COUNT
C      TICKS:  DEFW     0      ;FREE RUNNING MILISECOND COUNTER
C      DRVTMR: DEFB     0      ;DISK ACTIVITY TIMER
C      ;
C      ;
C      ;
C      IOBUFF EQU      0C100H      ;ATARI I/O BUFFER
C      LEN     EQU      512
C      ;
C      TRKBUF EQU      IOBUFF+LEN      ;TRACK BUFFER FOR READS
C      ;
C      ;
C      .DEPHASE
C      ;
C      LAST   EQU      8
C      END

```

FFB6
FFB6
FFB8
FFBA
FFBC
FFBD
FFBE

000A

FFC0
FFC2
FFC4

FFC6
FFC7
FFC8
FFCA

C100
0200

C300

1073

SERDRV.BAS: a Serial Printer Driver for ATARI DOS and OS/A+

A serial printer can be connected to the RS-232 Port of the ATR3000. J12, the jumper option that configures this port, is factory set to support an Epson serial printer. If you use a different type of serial printer, consult the printer's manual and Section II.6 to properly configure J12.

Serial printer drivers for printing under CP/M are explained in the ATR3000 CP/M Supplement. The following serial printer driver is for serial printing under ATARI DOS or OS/A+.

SERDRV.BAS is a dtrline printer driver. To work with this driver, your printer needs to use Pin 20 for hard wire handshaking. The driver is set to have no parity and a word length of 8 bits. You may choose a baud rate of 1200, 2400, 4800 or 9600.

Enter this program in ATARI BASIC, then run it before printing (if the computer is rebooted or reset, the program will have to be run again). When the program has been loaded, the message SERIAL DRIVER INSTALLED will show on the screen.

CAUTION: If a parallel printer is also used on a system where a serial printer is being used from the RS-232 Port, once SERDRV.BAS has been run, the computer will have to be rebooted or reset before printing can be done from the parallel PRINTER Port.

SERDRV.BAS

```
50 REM <<SERDRV.BAS>>
55 REM 8 BITS,NO PARITY,1200 BAUD
100 REM <<SERIAL PRINTER DRIVER>>
110 DIM BUFS(256)
120 Z=ADR("  "):REM TYPE 4 SPACES
130 DATA 104,76,89,228
140 FOR I=0 TO 3
150 READ X
160 POKE Z+I,X
170 NEXT I
180 DCB=768
190 DATA 90,1,83,0,0,0,1,0,6,0,18,240
200 FOR I=0 TO 11
210 READ X
220 POKE DCB+I,X
230 NEXT I
240 X=USR(Z)
250 A=ADR(BUFS)
260 X=INT(A/256)
270 POKE DCB+5,X
280 POKE DCB+4,A-X*256
290 DATA 195,0,239,195,25,239
300 FOR I=0 TO 5
310 READ X
320 POKE A+I,X
330 NEXT I
```

```

340 POKE DCB+2,ASC("W")
350 POKE DCB+3,128
360 POKE DCB+10,6
370 X=USR(Z)
380 POKE DCB+2,ASC("S")
390 POKE DCB+3,0
400 POKE DCB+10,0
410 POKE DCB+11,239
420 X=USR(Z)
430 GOSUB 2000
440 POKE DCB+2,ASC("W")
450 POKE DCB+3,128
460 POKE DCB+8,31
461 REM NUMBER OF PROGRAM BYTES
470 POKE DCB+10,3:
480 X=USR(Z)
490 PRINT "SERIAL DRIVER INSTALLED"
500 END
2000 DATA 243,6,9,175,205,16,239
2001 DATA 55,121,205,16,239,16
2002 DATA 251,251,201,211,81,31
2003 DATA 22,200,21,32,253,201
2004 DATA 219,80,0,230,2,201
2010 FOR I=0 TO 30
2020 READ X
2030 POKE A+I,X
2040 NEXT I
2050 RETURN

```

Baud rate is determined by the second variable in Line 2003. The above listing is for a baud rate of 1200. These are the possibilities:

1200 Baud:	2003 DATA 22,200,21,32,253,201
2400 Baud:	2003 DATA 22,100,21,32,253,201
4800 Baud:	2003 DATA 22,48,21,32,253,201
9600 Baud:	2003 DATA 22,22,21,32,253,201

ADDENDUM: The Terminal Program

The 40 column display window of the terminal program (Pages 59 - 60) has been updated to include more commands for changing the window's display. The keyboard commands are direct commands which are not transmitted to the ATR8000. The other commands must be transmitted from the ATR8000.

Moving the Display Window:

From the Keyboard

- 1) CTRL BACK S This toggle command switches the display window to show either the first 40 columns or the last 40 columns of the 80 column display. For example, if the first 40 columns show on the screen (Columns 0 - 39), entering CTRL BACK S changes the display to show the last 40 columns (Columns 40 - 79). [A HOME code also moves the window to show Columns 0 - 39.]
- 2) CTRL ; Moves the display window one column to the right. (The window will not move if this command is entered and the last 40 columns are currently displayed on the screen {Columns 40 - 79}.)
- CTRL . Moves the display window one column to the left. (The window will not move if this command is entered and the first 40 columns are currently displayed on the screen {Columns 0 - 39}.)
- 3) CTRL 3 This toggle command turns the Auto Display mode on and off. When the Auto Display mode is active, the display window moves with the cursor. (The default is on.)

From the ATR8000

- 1) ESC > An ESC > sequence moves the display window one column to the right. (The window will not move if this command is received and the window currently shows the last 40 columns {Columns 40 - 79}.)
- ESC < An ESC < sequence moves the display window one column to the left. (The window will not move if this command is received and the screen currently displays the first 40 columns {Columns 0 - 39}.)
- 2) ESC Zn An ESC Z sequence can be used to set the Auto Display mode to follow the cursor. For more information about ESC Z and the ESC Z variables, consult Pages 64 - 65.

Interpreting CP/M Error Messages: The Status Register

The Status Register monitors various conditions in the floppy disk controller (part of the circuitry of the ATR8000's board). This register is made up of 8 bits, with each individual status bit having its own meaning.

When a CP/M error message appears on the screen, it may be followed by a number. This number refers to one or more of the status bits. [Although some of the status bits point out errors, not all do. Some simply tell which conditions in the floppy disk controller are active.] The Status Register chart below shows the individual bits (Bits 0 - 7). The decimal equivalent of each bit is also referenced because the SWP BIOS uses decimal output for error messages.

To use the following chart, first determine the type of command that was active when the error message occurred (whether the floppy disk controller was trying to seek, read, or write). Then cross reference this command with the decimal number that appeared in the error message.

For example, ERROR 64, when the system was trying to write to a disk, means that there is a write protect error. This means that you were attempting to write to a disk that was write protected (for example, if an 8" disk is involved, you need to place a write-protect tab on it, a 5 1/4" disk needs the tab removed).

If a number in an error message is not one of the 8 decimal numbers above (1, 2, 4, 8, 16, 32, 64, or 128), then it is a combination of two. For example, ERROR 24 means that there is a problem with Bit 4 (16 decimal) and with Bit 3 (8 decimal).

More detailed explanation of the bit meanings follow the chart.

A Summary of the Status Register

COMMAND ↓	STATUS BITS							
	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	Decimal 128	Decimal 64	Decimal 32	Decimal 16	Decimal 8	Decimal 4	Decimal 2	Decimal 1
Seek	NOT READY	WRITE PROTECT	HEAD LOADED	SEEK ERROR	CRC ERROR	TRACK 0	INDEX	BUSY
Read	NOT READY	-	RECORD TYPE	RECORD NOT FOUND	CRC ERROR	LOST DATA	DATA REQUEST	BUSY
Write	NOT READY	WRITE PROTECT	WRITE FAULT	RECORD NOT FOUND	CRC ERROR	LOST DATA	DATA REQUEST	BUSY

NOT READY indicates that the requested drive is not ready. It can occur if the drive door is not closed, or if you inadvertently try to access a drive that doesn't exist.

WRITE PROTECT means that the write protect feature is active and the disk cannot be written to. To write to an 8" disk, you must place a foil write-protect tab on the disk. To write to a 5 1/4" disk, you must remove the write-protect tab from the disk.

HEAD LOADED means that the drive head is loaded and engaged.

RECORD TYPE. When a record is being read, this indicates the record-type code from the data field address mark. 1= deleted data mark. 0= normal data mark.

WRITE FAULT indicates that there is a fault in writing.

SEEK ERROR means that the desired track was not verified.

RECORD NOT FOUND means that the desired track, sector, or side was not found.

CRC ERROR means that there was a checksum error in data in a sector. If this occurs with a RECORD NOT FOUND or a SEEK error (decimal 24), then there is an error in one or more ID fields. If it occurs alone, it indicates error in a data field.

TRACK 0 indicates that the drive's Read/Write head is positioned to Track 0.

LOST DATA occurs when the computer did not respond to a data request in one byte time. This happens when you try to read a longer sector than the active program was expecting.

INDEX means that the index mark is detected from the drive.

DATA REQUEST is a copy of the data request output. It indicates that the data request is full on a read operation, or that the data request is empty on a write operation.

BUSY means that a command is under execution.

Addendum: Additions to DDSYSGEN.COM

DDSYSGEN now has an additional option that allows the user to customize the double density CP/M system tracks. Head load, timing, head settling time, drive step rate, parameters for the RS-232 Port (including baud rate, parity and word length), selections necessary for connecting particular printers to the ATR8000, number of drives used and type of drives used are some of values that can be altered.

The ATR8000 system allows the connection of a wide variety of peripherals. As different peripherals need different stipulations in the system tracks, these tracks are preconfigured for a particular set. The following set of peripherals is the SWP standard:

- single-sided, 40 track 5 1/4" drives.
- single-sided, 77 track 8" drives.
- 16 ms head settling time.
- a step rate that automatically adjusts to the fastest rate the drive is capable of running. (This rate can be marginal for some drives, so the symbolic patch function allows the user to set a defined rate.)
- a 50 ms head load delay.
- 1k byte block size for 5 1/4" disks.
- a parallel printer.
- RS-232 communications port configured for 1200 baud/odd parity/7 bits/full duplex.

If you have other peripherals, or if your peripherals do not meet all of the above standards, the CP/M system tracks must be customized using the third option of DDSYSGEN.

The third option of DDSYSGEN, Generate Custom system, first reads the double density CP/M tracks into memory, then allows changes to be made using the symbolic patch function.

After selecting #3 from the DDSYSGEN menu, the screen will tell you "insert disk with SYSTEM.SWP file into drive A and press <RETURN>." This file contains the double density CP/M system tracks; it comes on the user's distribution CP/M disk.

The screen will tell you that the computer is reading the system. If SYSTEM.SWP was not on the disk in Drive A when <RETURN> was pressed, the error message "CANNOT READ SYSTEM FILE" will show on the screen along with the message "press any key to return to main menu." Pressing any key will return the system to the DDSYSGEN menu.

After SYSTEM.SWP has been read, the question "Do you wish to use the symbolic patch function (Y/N)?" will appear on the screen. To customize the system tracks, enter Y. The screen now asks for the location to change (locations are listed in the following subsections). After entering a location, the computer shows the default value of the location and allows a change to be entered.

Symbolic patch function data entries are entered in hexadecimal. Binary to hex, decimal to hex, and decimal to binary charts follow. Use these charts along with the instructions in each subsection to compute proper data entries for any changes made.

<u>Binary</u>	<u>Hex</u>	<u>Decimal</u>	<u>Hex</u>	<u>Decimal</u>	<u>Hex</u>	<u>Decimal</u>	<u>Hex</u>
0000	0	0	0	50	32	100	64
0001	1	1	1	51	33	101	65
0010	2	2	2	52	34	102	66
0011	3	3	3	53	35	103	67
0100	4	4	4	54	36	104	68
0101	5	5	5	55	37	105	69
0110	6	6	6	56	38	106	6A
0111	7	7	7	57	39	107	6B
1000	8	8	8	58	3A	108	6C
1001	9	9	9	59	3B	109	6D
1010	A	10	A	60	3C	110	6E
1011	B	11	B	61	3D	111	6F
1100	C	12	C	62	3E	112	70
1101	D	13	D	63	3F	113	71
1110	E	14	E	64	40	114	72
1111	F	15	F	65	41	115	73
		16	10	66	42	116	74
		17	11	67	43	117	75
		18	12	68	44	118	76
		19	13	69	45	119	77
		20	14	70	46	120	78
		21	15	71	47	121	79
		22	16	72	48	122	7A
		23	17	73	49	123	7B
		24	18	74	4A	124	7C
		25	19	75	4B	125	7D
		26	1A	76	4C	126	7E
		27	1B	77	4D	127	7F
		28	1C	78	4E	128	80
		29	1D	79	4F	129	81
		30	1E	80	50	130	82
		31	1F	81	51	131	83
		32	20	82	52	132	84
		33	21	83	53	133	85
		34	22	84	54	134	86
		35	23	85	55	135	87
		36	24	86	56	136	88
		37	25	87	57	137	89
		38	26	88	58	138	8A
		39	27	89	59	139	8B
		40	28	90	5A	140	8C
		41	29	91	5B	141	8D
		42	2A	92	5C	142	8E
		43	2B	93	5D	143	8F
		44	2C	94	5E	144	90
		45	2D	95	5F	145	91
		46	2E	96	60	146	92
		47	2F	97	61	147	93
		48	30	98	62	148	94
		49	31	99	63		95

<u>Decimal</u>	<u>Binary</u>
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000
17	10001
18	10010
19	10011
20	10100
21	10101
22	10110
23	10111
24	11000
25	11001
26	11010
27	11011
28	11100
29	11101
30	11110
31	11111

10. [Notice that you do NOT add the digits together (i.e., 1+0=1), rather the left four bits make up the left digit, 1, and the right four bits make up the right digit, 0.]

If you change any of these locations, be sure that RATEA is changed for Drive A, RATEB for Drive B, and so on.

Head Load Delay

Location: HDLDTM

Head load delay is the amount of time between selecting a new drive and doing any reads or writes to the new drive. The value is determined in decimal, then converted to hex. The default is 50 ms, or 32 hex. Possible decimal values are from 1 to 255 ms.

Multiple or Single Drive System

Location: ONEDSK

The value at this location determines if the system is a single drive or a multiple drive system. The default is for multiple drives. Single drive system users need to change this value.

Value	Meaning
0	Multiple drives
1	Single drive

Type of Drives

Locations: NTRKSA, NTRKSB
NTRKSC, NTRKSD

These locations correspond to Drive A, Drive B, Drive C and Drive D. Enter the appropriate hex value for each drive using the following chart:

Value (hex)	Type of Drive
23	35 track, single-sided 5 1/4" drive
9F	35 track, double-sided 5 1/4" drive
28	*40 track, single-sided 5 1/4" drive
A8	40 track, double-sided 5 1/4" drive
50	80 track, single-sided 5 1/4" drive
D0	80 track, double-sided 5 1/4" drive
4D	*77 track, single-sided 8" drive
CD	77 track, double-sided 8" drive

* The ATR8000 is preconfigured to accept these types of drives, so it is not necessary to enter either of these values. (It can distinguish between 5 1/4" and 8" drives.)

For users with nonstandard drives:

If your drive is not listed above, the value can be determined by the following process.

The value can be determined in decimal, then entered in hex.

Single-sided drives:

Determine the number of tracks the drive has. Then use the decimal to hex charts to convert this number to hex. For example, 40 decimal equals 28 hex. Enter this value for the location.

Double-sided drives:

Determine the number of tracks the drive has. Now add 128 (decimal) to the track number. Convert this number to hex and enter it at the location.

For example: To compute the value for a 40 track, double-sided drive, add 40 and 128. 168 = A8 hex, the value to be entered.

CP/M Disk Parameter Blocks

Locations: DD8IN, DD5IN

The CP/M block size is preset to be 2k byte blocks for 8" disks and 1k byte blocks for 5 1/4" disks. WE DO NOT RECOMMEND CHANGING THE 8" BLOCK SIZE.

5 1/4" Users: CP/M only allows for 256 1k blocks of disk storage. Double-sided 5 1/4" drives and single-sided 80 track drives exceed this (remember this is double density CP/M -- a double density, double-sided 40 track drive can have up to 390k bytes of storage). In order to realize all possible space, the block size needs to be changed to 2k blocks.

If any double-sided 5 1/4" drives or any 80 track drives are connected, make the following location changes:

<u>Location</u>	<u>Default</u>	<u>Enter</u>
*DD5IN+4	03	04
DD5IN+5	07	0F
DD5IN+9	3F	7F
DD5IN+13	10	20

*To access the location +n, first enter the location (DD5IN), then press <RETURN> until the proper n value is accessed.

RS-232 Port Configuration

Locations: IBAUD, IBAUD+1

The RS-232 Port is preconfigured to be 1200 baud, odd parity, 7 bits and full duplex. These parameters must match the parameters of the peripheral interfaced to the RS-232 Port (and J12 must be properly configured).

If the baud rate in IBAUD is changed, IBAUD+1 must also be changed.

IBAUD:

This location's value is determined by computing the values in a byte (8 bits). The numbers are entered in binary, then converted